B-020

UI設計ツールを用いた開発における効率的運用

Efficient Cooperative Development Based on UI Design Tool

岡本 啓嗣† 中川 隆志† 小中 裕喜†
Hirotsugu Okamoto Takashi Nakagawa Hiroki Konaka

1．はじめに

近年、家電機器など様々な組み込み機器における User Interface (UI)の高度化、開発期間の短縮にともない、組み込み S/W 開発における UI 開発の生産性向上が重要な課題となっている。筆者らは、状態遷移図によって UI 部品設計を行い、その成果物を元にコード生成することで S/W 開発効率化を図る、UI 設計ツール Edamame (Embedded system Design Architecture with Model-based Approach and Middle-ware Environment)の開発を行ってきた[1][2][3]。Edamame では、設計された UI 部品を複数の画面中に配置して再利用することで、開発効率向上を図ることができる。またその設計を、画面デザインと制御ロジックとに分離し、それぞれの設計者がスキルに応じて作業することを可能にするとともに、分業化により同時並行的に開発を行うことができる[3]。

上記特長を活かすためには、Edamame による開発の運用面において、再利用の頻度を考慮しながら UI 部品を設計すること、及び各 UI 部品設計において適切に作業分担することが重要となる。本稿では、Edamame で UI 部品を設計する際に用いられる設計モデルについて説明し、開発における UI 部品の分類の方針と、各 UI 部品設計時の分業体制について述べる。

2．UI設計ツール Edamame 概要

2.1 SCO (State Chart Object)

Edamame は、表示の切り替えを伴う UI の階層的モデリングに適した、SCO (State Chart Object) という概念をベースとする。SCO とは、複数の状態とそれらの間の遷移を設計することが可能な UI 部品である。図１のように、ある SCO を別の SCO 中の画面に UI 部品として配置することができるため、画面中のカスタム表示部品からアプリケーションまで、様々な階層の設計を部品化し、組み合わせて設計することを可能としている。SCO は複数の画面に配置可能であり、またその仕様変更や機能改善は、自動的にそれらの配置画面に反映されるため、再利用による開発効率向上を可能としている。

2.2 スキルに応じた分業化

上記 SCO の設計には大きく分けて、画面デザインと制御ロジック設計という二つの作業が存在し、それぞれ異なる種類のスキルが必要となる。Edamame では、これらの作業の成果物が分離されており、それぞれに応じたスキルを持つ作業者が、同時並行的に開発を進めることが可能である。以降では、画面デザインの設計者を画面デザイナ、制御ロジックの設計者をプログラマと呼ぶ。

2.3 分業のための設計モデル

図１の SCO に対応した Edamame の設計モデルは、図２のようになる。設計モデルは、以下のように分割される。

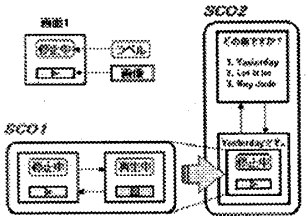

図１：SCO (State Chart Object) の概念

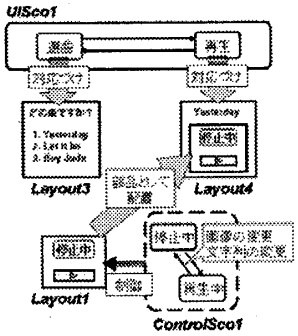

図２：Edamame における設計モデルの例

・Layout：画面デザインの設計モデル。UI 部品の配置などを行う。

・UISco：画面間遷移制御ロジックの設計モデル。画面の移り変わりを設計するために用いられ、複数の状態とそれらの間の状態遷移を持つ。また、各状態に対応する表示として Layout を持つ。

・ControlSco：画面内遷移制御ロジックの設計モデル。Layout の表示状態の制御に用いられる。UISco とは異なり、各状態に対応する Layout を持つのではなく、CcontrolSco 全体が Layout に対応づけられ、各状態は画面の内部状態を表している。図２では ControlSco1 が Layout1 の表示状態を制御しており、状態に応じて画像を変更する等の処理を行う。

Layout および UISco が UI 部品として再利用でき、図２では Layout1 が他の画面中に部品として配置されている。ControlSco と Layout の対応は、部品として配置されても維持されている。そのため、Layout が配置されているどの画面においても同様の制御が行われる。Layout1+ControlSco1 の部分を UISco で設計することも可能であるが、各状態に対応する画面に配置される部品は同一であるため、UISco の各状態に対する複数の Layout を設計するよりも、単一の Layout を ControlSco で制御するほうが、画面設計の共通化を図ることができる。

このようなモデルとすることで、画面デザイナが Layout を設計し、プログラマが ControlSco や UISco を設計してその制御を行うという、それぞれのスキルを活かした分業が可能である。

†三菱電機株式会社 先端技術総合研究所
Advanced Technology R&D Center, Mitsubishi Electric Corp.

4. 再利用頻度に応じた分類と分業方法

4.1 UI 部品の階層的分類

設計の再利用性を高めるためには、アプリケーション全体から、UI 部品として切り出すべき部分を適切に決定する必要がある。ここでは、出現の頻度に応じた以下の４種類に分類に基づき、UI 部品を切り出して設計することを想定している。

１．小部品：ボタンのように基本的な機能を提供する UI 部品。アプリケーション全体で頻繁に出現する。

２．中部品：ボタンが縦や横に並ぶリストのような、アプリケーション中のある範囲において共通的に利用される UI 部品。小部品ほどではないものの、しばしば出現する。

３．画面部品：待ち受け画面、メニュー画面といった、アプリケーションレベルでの画面を表す UI 部品。出現頻度は低い。

４．アプリケーション：アプリケーション全体における画面フロー。画面部品がどのような順序で表示されるかを設計する。基本的にアプリケーション全体で一つしか存在しない。

このように、出現頻度に応じて UI 部品を切り出すことで、自ずと再利用が促進される。

4.2 各階層における設計の分担

上記のように分類された UI 部品を、画面デザイナとプログラマで分担して設計するが、これを Edamame における設計データに照らすと、図３のように階層化、分業化される。トップレベルのアプリケーションにおける画面遷移を UISco で設計し、そこに至るまでの各階層の部品は、Layout+ControlSco で設計する。図中点線で区切っているように、中部品、画面部品の Layout を画面デザイナが担当し、それ以外はプログラマが担当する。以下、各階層について具体的に説明する。

小部品であるボタン部品は、Layout としては画像とラベル程度の部品が配置されるのみで単純であるため、必ずしも画面デザイナが Layout 設計に携わる必要がない。また、多くの箇所で利用されるため、配置箇所に応じて、表示データを上位から設定して利用できるようにしたり、動作にもバリエーションを持たせたりするなど、ある程度の汎用性を考慮して設計しておく必要があり、難易度が高い。そのため、小部品は比較的高いプログラミングスキルを持つプログラマが、Layout の設計と、ControlSco からの制御の両方を駆使しながら、完成度の高い部品を設計する。ここでは例えばボタンが押されたときにハイライト状態の画像に変更し、離されたときには通常の画像に変更するといった動作の設計や、ラベルの文字列を外部から設定できるように公開するといった設計を行う。

中部品、画面部品になると、配置される部品が増えてくるため、画面デザイナが Layout を担当する。例えば中部品であるリストは、小部品のボタンを並べて設計する。画面部品である音楽メニューでは、中部品のリストを利用して設計している。それらを制御するロジック部分である ControlSco は、プログラマが担当する。リストのような中部品に対応した ControlSco のロジックでは、上下キーが押されたときのフォーカス制御のような動作の設計が必要であり、ある程度高度なプログラミングスキルを求められる。一方、画面部品では、開発機種の仕様に関する知識が要求

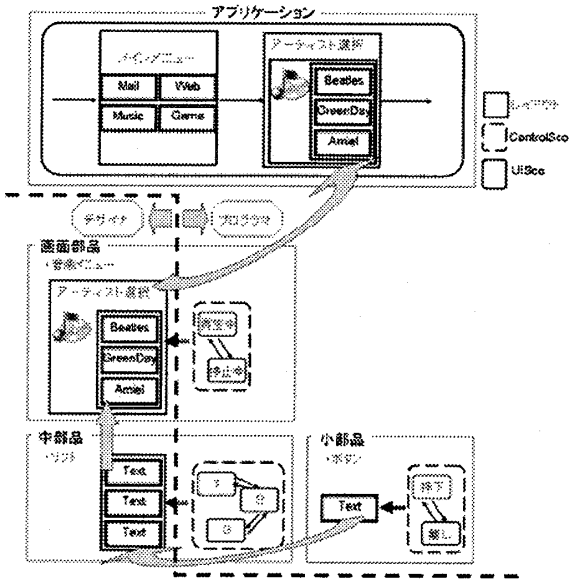

図３：階層化と分業

される。音楽メニューでは、例えばシステムから上がってきた再生状態のイベントに対し、再生中か停止中かでアイコンを変化させたり、画面に応じた表示内容を下位の部品にセットする、といった設計を行う。

最上位階層に位置するアプリケーションの制御では、例えばメニューにおいて Music ボタンが選択されると、次の音楽メニュー画面に遷移するといった設計を、UISco を用いて行う。UISco の各状態に、画面部品である音楽メニュー等が対応づけられる。ここでは、開発機種における画面フローを正確に把握することが必要である。

小部品、中部品、画面部品は Layout+ControlSco で設計しているが、これに限定されるものではない。表示内容がまるで異なる状態遷移を起こすような場合、UISco を用いて設計するなど、部品の特性に応じて設計モデルを選択することも可能である。

このように UI 部品を分類し、それぞれの設計作業をスキルの種類や知識に応じて分業することで、設計者のスキルを活かした再利用性の高い設計を行うことができる。

5. おわりに

本稿では、SCO をベースとした UI 設計ツール Edamame における、UI 部品の再利用とスキルに応じた分業を適切に実施していくための設計モデルについて述べた。また、再利用頻度を考慮して UI 部品を階層的に分類する方針と、各階層における設計の分担方法について述べた。

今後は本方針に基づいて、様々な機器開発へ本ツールを展開していく予定である。

参考文献

[1] 小中裕喜, ほか：階層的部品定義に基づく組み込み用 UI 設計ツール, 組み込みソフトウェア工学シンポジウム 2002, 情報処理学会研究報告, 2002-SE-139, 7-8 (2002)

[2] 小中裕喜, 中川隆志, ほか：組み込み用 UI 設計ツールをベースとした UI 設計開発, 三菱電機技報, Vol.77, No.7 (2003)

[3] 岡本啓嗣, 中川隆志, 小中裕喜：スキルに応じて作業分担可能な UI 設計ツール, 電気学会全国大会, Vol3, pp154, 155 (2007)